REMARKS/ARGUMENTS

35 U.S.C. §103

Claims 98 – 104 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent. No. 6,038,563 ("Bapat") in view of several combinations as proposed by the Examiner. Claims 98 and 101 stand rejected as being unpatentable over Bapat in view of U.S. Patent. No. 5,956,715 ("Glasser"). Claim 99 stands rejected over Bapat in view of Glasser in further view of U.S. Patent. No. 6,298,445 ("Shostack"). Claim 100 stands rejected over Bapat in view of Glasser in further view of U.S. Patent. No. 6,321,337 ("Reshef"). Claims 102 – 104 stands rejected over Bapat in view of Glasser in further view of U.S. Patent. No. 6,405,318 ("Rowland").

As a preliminary matter, Applicant requests clarification as to the obviousness rejections for claims 99, 100 and 102 – 104. In each rejection, the Examiner combines Bapat and Glasser with a different reference in an attempt to cure the admitted deficiencies of Bapat. Applicant submits the rejections of claims 99, 100 and 102 – 104 reveal a pattern of improper reliance on the prior art resulting in the same erroneous determinations of obviousness. More specifically, the Examiner relies on Bapat to teach or suggest the "preventing" steps as recited in the noted claims. Indeed, the Examiner relies on the same exact portions of Bapat in support of rejecting each of the "preventing" steps regardless of what Applicant is claiming to prevent.

For claim 99, the Examiner acknowledges Bapat's failure to disclose "processing the plurality of database events by detecting whether an executable SQL statement exploits a buffer overflow vulnerability in the database." The Examiner then asserts an obvious combination with Shostack to cure this deficiency. In the combination as proposed, the

Examiner improperly relies on Bapat to teach or suggest the step of preventing "**the** executable SQL statement from executing." In support, the Examiner cites to Bapat's general discussion of applying access control rules to requests. (Bapat, Col. 12:19 – 26).

Applicant respectfully traverses this reliance on Bapat as the Examiner **expressly states** Bapat fails to disclose the same executable SQL statement (one that exploits a buffer overflow vulnerability) as recited in the determining step of claim 99. Indeed, Bapat fails to even mention any concern for buffer overflow vulnerability, let alone prevent a SQL statement that exploits the buffer vulnerability from executing. Since Bapat fails to disclose determining whether the SQL statement exploits buffer overflow vulnerability, it is impossible for Bapat to prevent this type of SQL statement. Therefore, Applicant respectfully requests the withdrawal of the Examiner's rejection of claim 99 for obviousness.

For claim 100, the Examiner acknowledges Bapat's failure to disclose "detecting whether an executable SQL statement includes an operating system call." The Examiner then asserts obvious combination with Reshef to cure this deficiency. In the combination as proposed, the Examiner improperly relies on Bapat to teach or suggest the step of preventing "the executable SQL statement from making the operating system call." In support, the Examiner cites to Bapat's general discussion of applying access control rules to requests. (Bapat, Col. 12:19 – 26).

Applicant respectfully traverses this reliance on Bapat as the Examiner **expressly states** Bapat fails to disclose the same executable SQL statement (one that includes an operating system call) as recited in the detecting step of claim 100. Indeed, Bapat does not mention an SQL statement including an operating system call, let alone prevent the SQL statement from

making the operating system call. Since Bapat fails to disclose detecting whether an executable SQL statement includes an operating system call, it is impossible for Bapat to prevent this type of SQL statement. Therefore, Applicant respectfully requests the withdrawal of the Examiner's rejection of claim 100 for obviousness.

For claim 102, the Examiner acknowledges Bapat's failure to disclose "wherein said unauthorized activity is interfering with auditing settings." The Examiner then asserts obvious combination with Rowland to cure this deficiency. In the combination as proposed, the Examiner improperly relies on Bapat to teach or suggest the step of preventing "the set of auditing configurations from being altered." In support, the Examiner cites to Bapat's general discussion of applying access control rules to requests. (Bapat, Col. 12:19 – 26).

Applicant respectfully traverses this reliance on Bapat as the Examiner **expressly** states Bapat fails to disclose the same executable SQL statement (one that alters a set of auditing configurations existing on the database) as recited in the determining step of claim 102. At best, Bapat uses a log server to keep a security audit trail in one of two modes, detailed and abbreviated. (Bapat, Col. 12:58 – Col. 13:12; Col. 16:55 – Col. 17:14). While Bapat may keep a log of access grant and access denial events, the log server does not teach or suggest preventing a change from one mode to another. As such, the Examiner's reliance on Bapat to teach or suggest preventing the interference with auditing settings as claimed is improper. For the sake of brevity, Applicant submits the same arguments are equally applicable to claims 103 and 104. Therefore, Applicant respectfully requests the withdrawal of the Examiner's rejection of claim 102 – 104 for obviousness.

In the interests of compact prosecution, Applicant submits the following amendments place the pending claims in condition for allowance. Without prejudice or disclaimer, Applicant amends claim 98 to recite the elements "wherein the collector agent includes a plurality of collector definitions, each one of the collector definitions being associated with a database instance" and transmitting a signal "wherein transmitting the signal to the console includes using a dispatcher agent connected to the console over a peer-to-peer channel and transmission of the signal is platform independent."

Applicant submits the prior art of record, either alone or in combination, fails to teach or suggest the plurality of collector definitions where each one of the collector definitions is associated with a database instance because each prior art reference focuses on a building or designing protection for a single database. In other words, the cited references do not teach or suggest a scalable design as claimed by Applicant's "collector definitions."

Applicant further submits the prior art of record, either alone or in combination, fails to teach or suggest transmitting a signal wherein transmitting the signal to the console includes using a dispatcher agent connected to the console over a peer-to-peer channel and wherein the transmission of the signal is platform independent because each prior art reference is rooted in the single database model running on the same platform. Applicant submits the cited references do not teach or suggest the platform independent transmission of signals because the cited references do not, *inter alia,* connect a console with a dispatcher agent over a peer-to-peer communication channel.

Conclusion

Applicant believes the above is fully responsive to the examiner's concerns. Applicant further submits the rejection of the pending claims should be considered as no longer tenable with respect to the amended claims and should be withdrawn. Should the Examiner consider necessary or desirable any formal changes anywhere in the specification, claims and/or drawing, then it is respectfully asked that such changes be made by Examiner's Amendment, if the Examiner feels this would facilitate passage of the case to issuance. Alternatively should the Examiner feel that a personal discussion might be helpful in advancing this case to allowance, he is invited to telephone the undersigned.

Respectfully submitted,

Law Offices of Peter S. Canelias April 23, 2010

By: 

Peter S. Canelias
Reg. No. 40,547
Law Offices of Peter S. Canelias
420 Lexington Avenue-Suite 300
New York, NY 10170
Tel: (212) 223-9654
Fax: (212) 223-9651